Study guide: Time-dependent problems and variational forms

Hans Petter Langtangen[1,2]

Center for Biomedical Computing, Simula Research Laboratory[1]

Department of Informatics, University of Oslo[2]

Oct 30, 2015

Time-dependent problems

- So far: used the finite element framework for discretizing in space
- What about $u_t = u_{xx} + f$?
  1. Use finite differences in time to obtain a set of recursive spatial problems
  2. Solve the spatial problems by the finite element method

Example: diffusion problem

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \alpha \nabla^2 u + f(\boldsymbol{x}, t), && \boldsymbol{x} \in \Omega, t \in (0, T] \\
u(\boldsymbol{x}, 0) &= I(\boldsymbol{x}), && \boldsymbol{x} \in \Omega \\
\frac{\partial u}{\partial n} &= 0, && \boldsymbol{x} \in \partial\Omega,\ t \in (0, T]
\end{aligned}
$$

A Forward Euler scheme; ideas

$$[D_t^+ u = \alpha\nabla^2 u + f]^n, \quad n = 1, 2, \ldots, N_t - 1$$

Solving wrt u^{n+1}:

$$u^{n+1} = u^n + \Delta t \left(\alpha\nabla^2 u^n + f(\boldsymbol{x}, t_n)\right)$$

- $u^n = \sum_j c_j^n \psi_j \in V$, $u^{n+1} = \sum_j c_j^{n+1} \psi_j \in V$
- Compute u^0 from I
- Compute u^{n+1} from u^n by solving the PDE for u^{n+1} at each time level

A Forward Euler scheme; stages in the discretization

- $u_{\mathrm{e}}(\boldsymbol{x}, t)$: exact solution of the PDE problem
- $u_{\mathrm{e}}^n(\boldsymbol{x})$: exact solution of time-discrete problem (after applying a finite difference scheme in time)
- $u_{\mathrm{e}}^n(\boldsymbol{x}) \approx u^n = \sum_{j\in\mathcal{I}_s} c_j^n\psi_j =$ solution of the time- and space-discrete problem (after applying a Galerkin method in space)

$$\frac{\partial u_{\mathrm{e}}}{\partial t} = \alpha\nabla^2 u_{\mathrm{e}} + f(\boldsymbol{x}, t)$$

$$u_{\mathrm{e}}^{n+1} = u_{\mathrm{e}}^n + \Delta t\left(\alpha\nabla^2 u_{\mathrm{e}}^n + f(\boldsymbol{x}, t_n)\right)$$

$$u_{\mathrm{e}}^n \approx u^n = \sum_{j=0}^{N} c_j^n\psi_j(\boldsymbol{x}),\quad u_{\mathrm{e}}^{n+1} \approx u^{n+1} = \sum_{j=0}^{N} c_j^{n+1}\psi_j(\boldsymbol{x})$$

$$R = u^{n+1} - u^n - \Delta t\left(\alpha\nabla^2 u^n + f(\boldsymbol{x}, t_n)\right)$$

A Forward Euler scheme; weighted residual (or Galerkin) principle

$$R = u^{n+1} - u^n - \Delta t \left(\alpha \nabla^2 u^n + f(\boldsymbol{x}, t_n)\right)$$

The weighted residual principle:

$$\int_\Omega Rw \,\mathrm{d}x = 0, \quad \forall w \in W$$

results in

$$\int_\Omega \left[u^{n+1} - u^n - \Delta t \left(\alpha \nabla^2 u^n + f(\boldsymbol{x}, t_n)\right)\right] w \,\mathrm{d}x = 0, \quad \forall w \in W$$

Galerkin: $W = V$, $w = v$

A Forward Euler scheme; integration by parts

Isolating the unknown u^{n+1} on the left-hand side:

$$\int_\Omega u^{n+1}\psi_i \,\mathrm{d}x = \int_\Omega \left[u^n + \Delta t\left(\alpha\nabla^2 u^n + f(\boldsymbol{x}, t_n)\right)\right] v \,\mathrm{d}x$$

Integration by parts of $\int \alpha(\nabla^2 u^n) v \,\mathrm{d}x$:

$$\int_\Omega \alpha(\nabla^2 u^n) v \,\mathrm{d}x = -\int_\Omega \alpha\nabla u^n \cdot \nabla v \,\mathrm{d}x + \underbrace{\int_{\partial\Omega} \alpha\frac{\partial u^n}{\partial n} v \,\mathrm{d}x}_{=0 \quad \Leftarrow \quad \partial u^n/\partial n = 0}$$

Variational form:

$$\int_\Omega u^{n+1} v \,\mathrm{d}x = \int_\Omega u^n v \,\mathrm{d}x - \Delta t\int_\Omega \alpha\nabla u^n \cdot \nabla v \,\mathrm{d}x + \Delta t\int_\Omega f^n v \,\mathrm{d}x, \quad \forall v \in V$$

New notation for the solution at the most recent time levels

- u and `u`: the spatial unknown function to be computed
- u_1 and `u_1`: the spatial function at the previous time level $t - \Delta t$
- u_2 and `u_2`: the spatial function at $t - 2\Delta t$
- This new notation gives close correspondence between code and math

$$\int_\Omega uv \,\mathrm{d}x = \int_\Omega u_1 v \,\mathrm{d}x - \Delta t \int_\Omega \alpha\nabla u_1 \cdot \nabla v \,\mathrm{d}x + \Delta t \int_\Omega f^n v \,\mathrm{d}x$$

or shorter

$$(u, v) = (u_1, v) - \Delta t(\alpha\nabla u_1, \nabla v) + \Delta t(f^n, v)$$

Deriving the linear systems

- $u = \sum_{j=0}^{N} c_j \psi_j(\boldsymbol{x})$
- $u_1 = \sum_{j=0}^{N} c_{1,j} \psi_j(\boldsymbol{x})$
- $\forall v \in V$: for $v = \psi_i$, $i = 0, \ldots, N$

Insert these in

$$(u, \psi_i) = (u_1, \psi_i) - \Delta t(\alpha \nabla u_1, \nabla \psi_i) + \Delta t(f^n, \psi_i)$$

and order terms as matrix-vector products ($i = 0, \ldots, N$):

$$\sum_{j=0}^{N} \underbrace{(\psi_i, \psi_j)}_{M_{i,j}} c_j = \sum_{j=0}^{N} \underbrace{(\psi_i, \psi_j)}_{M_{i,j}} c_{1,j} - \Delta t \sum_{j=0}^{N} \underbrace{(\nabla \psi_i, \alpha \nabla \psi_j)}_{K_{i,j}} c_{1,j} + \Delta t(f^n, \psi_i)$$

Structure of the linear systems

$$Mc = Mc_1 - \Delta t K c_1 + \Delta t f$$

$$
\begin{aligned}
M &= \{M_{i,j}\}, \quad M_{i,j} = (\psi_i, \psi_j), \quad i,j \in \mathcal{I}_s \\
K &= \{K_{i,j}\}, \quad K_{i,j} = (\nabla\psi_i, \alpha\nabla\psi_j), \quad i,j \in \mathcal{I}_s \\
f &= \{(f(\boldsymbol{x}, t_n), \psi_i)\}_{i\in\mathcal{I}_s} \\
c &= \{c_i\}_{i\in\mathcal{I}_s} \\
c_1 &= \{c_{1,i}\}_{i\in\mathcal{I}_s}
\end{aligned}
$$

Computational algorithm

1. Compute M and K.
2. Initialize u^0 by either interpolation or projection
3. For $n = 1, 2, \ldots, N_t$:
   1. compute $b = Mc_1 - \Delta t K c_1 + \Delta t f$
   2. solve $Mc = b$
   3. set $c_1 = c$

Initial condition:

- Either interpolation: $c_{1,j} = I(\boldsymbol{x}_j)$ (finite elements)
- Or projection: solve $\sum_j M_{i,j} c_{1,j} = (I, \psi_i)$, $i \in \mathcal{I}_s$

Example using sinusoidal basis functions

$$\frac{\partial u}{\partial t} = \alpha\frac{\partial^2 u}{\partial x^2}, \qquad x \in (0, L),\ \ t \in (0, T], \tag{1}$$

$$u(x, 0) = A\cos(\pi \boldsymbol{x}/L) + B\cos(10\pi x/L), \qquad x \in [0, L], \tag{2}$$

$$\frac{\partial u}{\partial x} = 0, \qquad x = 0, L,\ \ t \in (0, T]\,. \tag{3}$$

$$\psi_i = \cos(i\pi x/L)\,.$$

Approximating the initial condition

$I(x) \in V$ implies perfect approximation of the initial condition:

$$c_{1,1} = A, \quad c_{1,10} = B,$$

while $c_{1,i} = 0$ for $i \neq 1, 10$.

Computing the M and K matrices

Note that ψ_i and ψ'_i are orthogonal on $[0, L]$ such that we only need to compute the diagonal elements $M_{i,i}$ and $K_{i,i}$!

$$M_{0,0} = L, \quad M_{i,i} = L/2,\ i > 0, \quad K_{0,0} = 0, \quad K_{i,i} = \frac{\pi^2 i^2}{2L},\ i > 0\,.$$

Solving the equation system

$$Lc_0 = Lc_{1,0} - \Delta t \cdot 0 \cdot c_{1,0},$$
$$\frac{L}{2}c_i = \frac{L}{2}c_{1,i} - \Delta t \frac{\pi^2 i^2}{2L} c_{1,i}, \quad i > 0\,.$$

$$c_i = (1 - \Delta t(\frac{\pi i}{L})^2)c_{1,i}\,.$$

We actually get a closed-form discrete solution:

$$u_i^n = A(1 - \Delta t(\frac{\pi}{L})^2)^n \cos(\pi x/L) + B(1 - \Delta t(\frac{10\pi}{L})^2)^n \cos(10\pi x/L)\,.$$

Comparing P1 elements with the finite difference method; ideas

- P1 elements in 1D
- Uniform mesh on $[0, L]$ with cell length h
- No Dirichlet conditions: $\psi_i = \varphi_i$, $i = 0, \ldots, N = N_n - 1$
- Have found formulas for M and K at the element level
- Have assembled the global matrices
- Have developed corresponding finite difference operator formulas
- M: $h[D_t^+(u + \frac{1}{6}h^2 D_x D_x u)]_i^n$
- K: $h[\alpha D_x D_x u]_i^n$

Comparing P1 elements with the finite difference method; results

Diffusion equation with finite elements is equivalent to

$$[D_t^+(u + \frac{1}{6}h^2 D_x D_x u) = \alpha D_x D_x u + f]_i^n$$

Can lump the mass matrix by Trapezoidal integration and get the standard finite difference scheme

$$[D_t^+ u = \alpha D_x D_x u + f]_i^n$$

Discretization in time by a Backward Euler scheme

Backward Euler scheme in time:

$$[D_t^- u = \alpha\nabla^2 u + f(\boldsymbol{x}, t)]^n$$

$$u_{\rm e}^n - \Delta t\left(\alpha\nabla^2 u_{\rm e}^n + f(\boldsymbol{x}, t_n)\right) = u_{\rm e}^{n-1}$$

$$u_{\rm e}^n \approx u^n = \sum_{j=0}^{N} c_j^n\psi_j(\boldsymbol{x}),\quad u_{\rm e}^{n+1} \approx u^{n+1} = \sum_{j=0}^{N} c_j^{n+1}\psi_j(\boldsymbol{x})$$

The variational form of the time-discrete problem

$$\int_\Omega \left(u^n v + \Delta t \alpha \nabla u^n \cdot \nabla v\right)\,\mathrm{d}x = \int_\Omega u^{n-1} v\,\mathrm{d}x + \Delta t \int_\Omega f^n v\,\mathrm{d}x, \quad \forall v \in V$$

or

$$(u, v) + \Delta t(\alpha \nabla u, \nabla v) = (u_1, v) + \Delta t(f^n, \psi_i)$$

The linear system: insert $u = \sum_j c_j \psi_i$ and $u_1 = \sum_j c_{1,j} \psi_i$,

$$(M + \Delta t K)c = Mc_1 + \Delta t f$$

Calculations with P1 elements in 1D

Can interpret the resulting equation system as

$$[D_t^-(u + \frac{1}{6}h^2 D_x D_x u) = \alpha D_x D_x u + f]_i^n$$

Lumped mass matrix (by Trapezoidal integration) gives a standard finite difference method:

$$[D_t^- u = \alpha D_x D_x u + f]_i^n$$

Dirichlet boundary conditions

Dirichlet condition at $x = 0$ and Neumann condition at $x = L$:

$$
\begin{aligned}
u(\boldsymbol{x}, t) &= u_0(\boldsymbol{x}, t), && \boldsymbol{x} \in \partial\Omega_D \\
-\alpha\frac{\partial}{\partial n}u(\boldsymbol{x}, t) &= g(\boldsymbol{x}, t), && \boldsymbol{x} \in \partial\Omega_N
\end{aligned}
$$

Forward Euler in time, Galerkin's method, and integration by parts:

$$
\int_\Omega u^{n+1} v \,\mathrm{d}x = \int_\Omega (u^n - \Delta t\alpha\nabla u^n\cdot\nabla v)\,\mathrm{d}x + \Delta t\int_\Omega fv\,\mathrm{d}x - \Delta t\int_{\partial\Omega_N} gv\,\mathrm{d}s,
$$

Requirement: $v = 0$ on $\partial\Omega_D$

Boundary function

$$u^n(\boldsymbol{x}) = u_0(\boldsymbol{x}, t_n) + \sum_{j\in\mathcal{I}_s} c_j^n \psi_j(\boldsymbol{x})$$

$$\sum_{j\in\mathcal{I}_s} \left(\int_\Omega \psi_i\psi_j \,\mathrm{d}x\right) c_j^{n+1} = \sum_{j\in\mathcal{I}_s} \left(\int_\Omega \left(\psi_i\psi_j - \Delta t\alpha\nabla\psi_i\cdot\nabla\psi_j\right) \mathrm{d}x\right) c_j^n -$$
$$\int_\Omega \left(u_0(\boldsymbol{x}, t_{n+1}) - u_0(\boldsymbol{x}, t_n) + \Delta t\alpha\nabla u_0(\boldsymbol{x}, t_n)\cdot\nabla\psi_i\right) \mathrm{d}x$$
$$+ \Delta t\int_\Omega f\psi_i \,\mathrm{d}x - \Delta t\int_{\partial\Omega_N} g\psi_i \,\mathrm{d}s, \quad i \in \mathcal{I}_s$$

Finite element basis functions

- $B(\boldsymbol{x}, t_n) = \sum_{j\in I_b} U_j^n \varphi_j$
- $\psi_i = \varphi_{\nu(j)}$, $j \in \mathcal{I}_s$
- $\nu(j)$, $j \in \mathcal{I}_s$, are the node numbers corresponding to all nodes without a Dirichlet condition

$$
\begin{aligned}
u^n &= \sum_{j\in I_b} U_j^n \varphi_j + \sum_{j\in\mathcal{I}_s} c_{1,j}\varphi_{\nu(j)},\\
u^{n+1} &= \sum_{j\in I_b} U_j^{n+1} \varphi_j + \sum_{j\in\mathcal{I}_s} c_j\varphi_{\nu(j)}
\end{aligned}
$$

$$
\begin{aligned}
\sum_{j\in\mathcal{I}_s}\left(\int_\Omega \varphi_i\varphi_j \,\mathrm{d}x\right) c_j &= \sum_{j\in\mathcal{I}_s}\left(\int_\Omega \left(\varphi_i\varphi_j - \Delta t\alpha\nabla\varphi_i\cdot\nabla\varphi_j\right)\mathrm{d}x\right) c_{1,j} -\\
&\quad \sum_{j\in I_b}\int_\Omega \left(\varphi_i\varphi_j(U_j^{n+1} - U_j^n) + \Delta t\alpha\nabla\varphi_i\cdot\nabla\varphi_j U_j^n\right)\mathrm{d}x\\
&\quad + \Delta t\int_\Omega f\varphi_i\,\mathrm{d}x - \Delta t\int_{\partial\Omega_N} g\varphi_i\,\mathrm{d}s,\quad i\in\mathcal{I}_s
\end{aligned}
$$

Modification of the linear system; the raw system

- Drop boundary function
- Compute as if there are not Dirichlet conditions
- Modify the linear system to incorporate Dirichlet conditions
- $\mathcal{I}_s$ holds the indices of all nodes $\{0, 1, \ldots, N = N_n - 1\}$

$$\sum_{j\in\mathcal{I}_s} \Big(\underbrace{\int_\Omega \varphi_i\varphi_j \,\mathrm{d}x}_{M_{i,j}}\Big) c_j = \sum_{j\in\mathcal{I}_s} \Big(\underbrace{\int_\Omega \varphi_i\varphi_j \,\mathrm{d}x}_{M_{i,j}} - \Delta t \underbrace{\int_\Omega \alpha\nabla\varphi_i\cdot\nabla\varphi_j \,\mathrm{d}x}_{K_{i,j}}\Big) c_{1,j}$$

$$\underbrace{+\Delta t\int_\Omega f\varphi_i \,\mathrm{d}x - \Delta t\int_{\partial\Omega_N} g\varphi_i \,\mathrm{d}s,}_{f_i} \quad i\in\mathcal{I}_s$$

Modification of the linear system; setting Dirichlet conditions

$$Mc = b, \quad b = Mc_1 - \Delta t K c_1 + \Delta t f$$

For each k where a Dirichlet condition applies, $u(x_k, t_{n+1}) = U_k^{n+1}$,

- set row k in M to zero and 1 on the diagonal: $M_{k,j} = 0$, $j \in \mathcal{I}_s$, $M_{k,k} = 1$
- $b_k = U_k^{n+1}$

Or apply the slightly more complicated modification which preserves symmetry of M

Modification of the linear system; Backward Euler example

Backward Euler discretization in time gives a more complicated coefficient matrix:

$$Ac = b, \quad A = M + \Delta t K, \quad b = Mc_1 + \Delta t f$$

- Set row k to zero and 1 on the diagonal: $M_{k,j} = 0$, $j \in \mathcal{I}_s$, $M_{k,k} = 1$
- Set row k to zero: $K_{k,j} = 0$, $j \in \mathcal{I}_s$
- $b_k = U_k^{n+1}$

Observe: $A_{k,k} = M_{k,k} + \Delta t K_{k,k} = 1 + 0$, so $c_k = U_k^{n+1}$

Analysis of the discrete equations

The diffusion equation $u_t = \alpha u_{xx}$ allows a (Fourier) wave component

$$u = A_{\rm e}^n e^{ikx}, \quad A_{\rm e} = e^{-\alpha k^2 \Delta t}$$

Numerical schemes often allow the similar solution

$$u_q^n = A^n e^{ikx}$$

- A: amplification factor to be computed
- How good is this A compared to the exact one?

Handy formulas

$$
\begin{aligned}
[D_t^+ A^n e^{ikq\Delta x}]^n &= A^n e^{ikq\Delta x}\frac{A-1}{\Delta t},\\
[D_t^- A^n e^{ikq\Delta x}]^n &= A^n e^{ikq\Delta x}\frac{1-A^{-1}}{\Delta t},\\
[D_t A^n e^{ikq\Delta x}]^{n+\frac{1}{2}} &= A^{n+\frac{1}{2}} e^{ikq\Delta x}\frac{A^{\frac{1}{2}}-A^{-\frac{1}{2}}}{\Delta t} = A^n e^{ikq\Delta x}\frac{A-1}{\Delta t},\\
[D_xD_x A^n e^{ikq\Delta x}]_q &= -A^n\frac{4}{\Delta x^2}\sin^2\left(\frac{k\Delta x}{2}\right)
\end{aligned}
$$

Amplification factor for the Forward Euler method; results

Introduce $p = k\Delta x/2$ and $C = \alpha\Delta t/\Delta x^2$:

$$A = 1 - 4C\frac{\sin^2 p}{\underbrace{1 - \frac{2}{3}\sin^2 p}_{\text{from } M}}$$

(See notes for details)

Stability: $|A| \leq 1$:

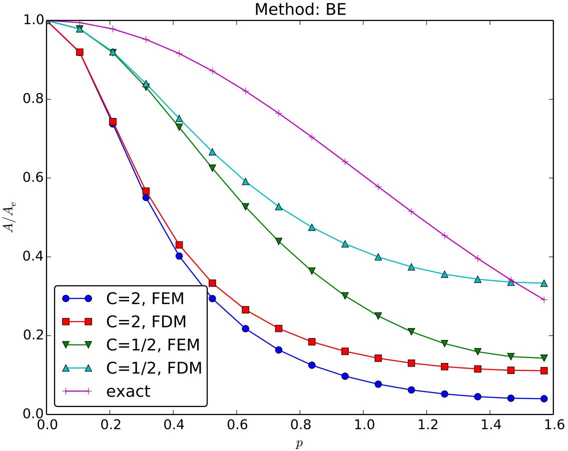
$$C \leq \frac{1}{6} \quad\Rightarrow\quad \Delta t \leq \frac{\Delta x^2}{6\alpha}$$

Finite differences: $C \leq \frac{1}{2}$, so finite elements give a *stricter* stability criterion for this PDE!

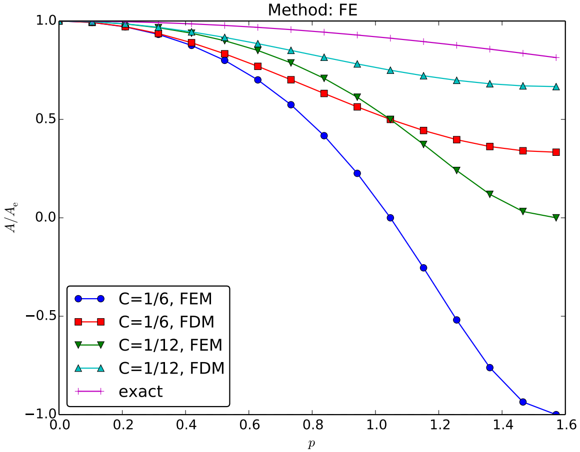
Amplification factor for the Backward Euler method; results

Coarse meshes:

$$A = \left(1 + 4C\frac{\sin^2 p}{1 + \frac{2}{3}\sin^2 p}\right)^{-1} \quad \text{(unconditionally stable)}$$



Amplification factors for smaller time steps; Forward Euler



Amplification factors for smaller time steps; Backward Euler